

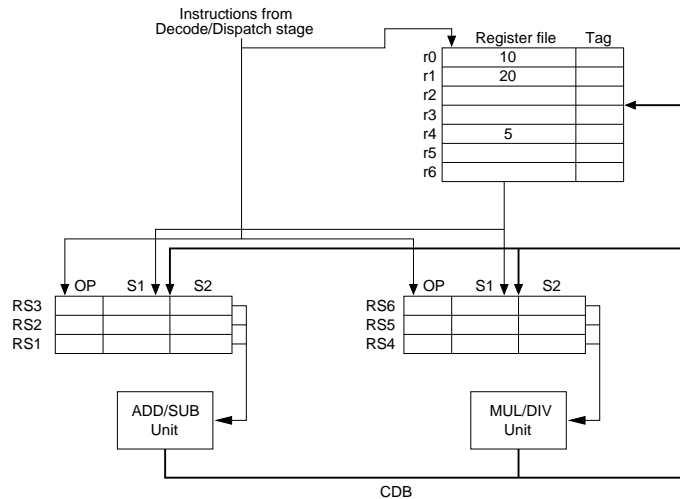
Tentamen Advanced Computer Architecture

Maandag 31 maart 2003
A.303, 9.30 - 12.30

Het is niet toegestaan bij het tentamen literatuur of aantekeningen te gebruiken. Geef altijd uitleg bij uw antwoorden. Succes!

- 1 a** U, als processor ontwerper, bent bezig om een processor te ontwerpen. U moet de blocksize van de datacache bepalen: 64 of 128 bytes. Na simulaties blijkt dat bij blokken van 64 bytes de cache een hit-ratio van 95% heeft en bij blokken van 128 bytes een hit-ratio van 98%. Tevens blijkt dat 25% van alle instructies geheugen accessen zijn. Een geheugen access resulterend in een cache hit kost 1 clock-cycle. Bij een cache miss met 64-byte blokken kost het 40 clock-cycli om een blok uit het main-memory te halen en bij 128-byte blokken kost het 72 cycli. Alle andere instructies kosten gemiddeld 1.2 clock-cycli. Welke keuze, een blocksize van 64 of 128 bytes, is beter?
- 1 b** Leg uit hoe een virtually indexed, physically tagged cache werkt. Leg tevens uit wat de "page colouring" techniek uitgevoerd door het Operating System te maken heeft met deze caches.
- 1 c** We weten dat caches gebruik maken van localiteit. Leg uit dat DRAMs dit ook doen.
-

Beschouw de volgende op Tomasulo scheduling gebaseerde architectuur (met register status):



Neem verder aan dat er per cycle twee instructies ge-dispatched kunnen worden: eentje naar het reservation station van de ADD/SUB unit en eentje naar die van de MUL/DIV unit. ADD/SUB instructies executeren in 1 cycle en MUL/DIV instructies in 5 cycles.

- 2 a** Laat voor de volgende 4 instructies zien hoe ze gescheduled c.q. ge-executeerd worden:

```
instructie i:    MUL R2, R0, R1    // R2 = R0*R1
instructie i+1: MUL R3, R2, R0    // R3 = R2*R0
instructie i+2: ADD R2, R4, R1    // R2 = R4+R1
instructie i+3: SUB R0, R3, R2    // R0 = R3-R2
```

Teken de status van de reservation stations en register file op de belangrijke cycles.

- 2 b** Leg uit welk probleem er ontstaat bij op Tomasulo scheduling gebaseerde architecturen wanneer er veel instructies *tegelijktijd* kunnen committen (na executie hun resultaten terug schrijven naar de architectuur registers).
 - 2 c** Beschrijf wat false dependencies zijn en laat zien – door middel van een voorbeeld – dat deze dependencies automatisch worden opgelost in Tomasulo scheduling.
-

- 3 a** Leg uit waarom het probleem van control hazards in de toekomst steeds erger wordt. Gebruik hierbij de aanname dat één op de vijf instructies een branch is.
 - 3 b** Leg uit wat een Trace Cache is en wat het voordeel is van een dergelijk cache ten opzichte van een traditionele instructie cache.
 - 3 c** Beschrijf het principe van een tournament branch predictor.
-

- 4 a** Leg uit waarom VLIW architecturen vaak geclusterd zijn. Leg ook wat voor invloed dit heeft op de compiler.
 - 4 b** Waarom wordt er veelvuldig gebruik gemaakt van code compressie binnen VLIWs en geef twee alternatieven voor de plaats waar decompressie kan plaatsvinden (geef ook de voor- en nadelen van beide alternatieven).
 - 4 c** Leg uit waarom klassieke VLIW processoren over het algemeen energie zuiniger zijn dan moderne superscalar processoren. Neem ook issues als clock frequentie en instructie-level parallelisme mee in je verklaring.
-

- 5 a** Geef een uitvoerige beschrijving van een aantal trends op het gebied van complexe (b.v. multimedia) embedded systemen. Leg ook uit wat de impact van deze trends op het ontwerp van deze systemen is.